



برنامه‌سازی پیشرفته

تمرین شماره‌ی ۷ - فاز ۲

مدرس: رامتین خسروی

طراحان: امین ستایش، علی زارع، آرمان رستمی، محسن فیاض

مهلت تحویل: سه‌شنبه ۹ دی، ساعت ۲۳:۵۵

Smart UTunes!

مقدمه



در فاز اول شما به پیاده‌سازی منطق برنامه و استفاده از رابط خط فرمان^۱ برای ارتباط با آن پرداختید. در این فاز امکاناتی جهت هوشمند کردن برنامه به آن اضافه می‌کنیم.

در این فاز پیاده‌سازی قابلیت‌هایی نظیر پیشنهاد آهنگ به افراد و اعلام درصد مشابهت با سایر کاربران انجام می‌شود تا بتوانیم به کاربران آهنگ‌های دیگری که ممکن است بپسندند را پیشنهاد دهیم. برای این کار از اطلاعاتی که در صفحه درس برای آهنگ‌های مورد علاقه خود ثبت کرده بودید استفاده می‌کنیم.

^۱ CLI (Command Line Interface)

شرح تمرین

در این فاز به پیاده‌سازی یک روش پیشنهاد دادن آهنگ‌ها به افراد بر اساس شباهت میپردازیم و چند دستور جدید به برنامه اضافه می‌کنیم تا بتوانیم از قابلیت‌های اضافه شده استفاده کنیم.

تمام نکاتی که برای دستورات در بخش قبل ذکر شده بود در این فاز نیز برقرار است و بایستی در برنامه به درستی بررسی شوند. توضیح خطاها مجدداً در انتهای صورت پروژه ذکر شده است که انتظار می‌رود به درستی بررسی شوند و پیغام مناسب را چاپ کنید.

خروجی پروژه شما مانند فاز قبل به صورت خودکار آزموده می‌شود، لذا خروجی شما باید دقیقاً همانند خروجی خواسته‌شده باشد. در غیر این صورت نمره‌ی بخش آزمون را از دست خواهید داد.

• آهنگ‌های پسندیده شده

در این فاز علاوه بر فایل CSV اطلاعات آهنگ‌ها، فایل CSV دیگری که شامل آهنگ‌های پسندیده کاربران و اطلاعات کاربران است قرار داده شده است که شامل آهنگ‌های موردعلاقه هر فرد می‌باشد که توضیح ستون‌های آن در ادامه آمده است.

دقت داشته باشید که اگر یک نفر چند آهنگ را پسندیده باشد برای او چند سطر در فایل CSV موجود است که پشت سر هم آمده‌اند.

اطلاعات ستون‌ها به شرح زیر است:

1. username که از ترکیب نام و نام خانوادگی شرکت‌کنندگان ایجاد شده است.
2. email نشانی پست الکترونیکی فرد می‌باشد.
3. password مجدداً از ترکیب نام و نام خانوادگی شرکت‌کنندگان ایجاد شده است.
4. liked_songs در این ستون شناسه آهنگ پسندیده شده توسط کاربر ذکر می‌شود.

• وارد کردن اطلاعات

اطلاعات دو فایل csv داده شده بایستی خوانده شده و اطلاعات کاربران و آهنگ‌های مورد پسند آن‌ها در حافظه برنامه ذخیره شود. دستور جدید به این شکل خواهد بود. **توجه کنید نام فایل اجرایی برنامه شما باید مجدداً `itunes.out` باشد. به ترتیب این فایل‌ها در آرگومان‌ها دقت کنید.**

ورودی

```
./itunes.out <songs_csv_file_relative_address>  
<liked_songs_csv_file_relative_address>
```

ورودی نمونه

```
./itunes.out folder1/folder2/songs.csv folder3/liked_songs.csv
```

سیستم پیشنهاددهنده آهنگ

برای پیاده‌سازی پیشنهاددهی² آهنگ‌ها مراحل زیر را طی می‌کنیم که در ادامه هر کدام با توضیحات بیشتر ذکر شده‌اند. در بخش توضیحات روند یک مثال هم توضیح داده شده‌است.

1. ایجاد ماتریس³ کاربران و آهنگ‌های پسندیده‌شده توسط آن‌ها.
2. محاسبه درصد مشابهت هر دو کاربر نسبت به یکدیگر.
3. میانگین وزن‌دار بر اساس درصد مشابهت یک کاربر به کاربران دیگر برای تعیین آهنگ‌های پیشنهادی.

² Recommendation

³ Matrix

○ ماتریس کاربران و آهنگ‌ها

در این مرحله با استفاده از اطلاعات آهنگ‌های پسندیده‌شده که در ابتدای برنامه در حافظه برنامه ذخیره کرده بودیم ماتریسی دوبعدی می‌سازیم که سطرهاى آن کاربران و ستون‌های آن آهنگ‌ها هستند. هر عنصر در این ماتریس می‌تواند یکی از دو مقدار ۰ یا ۱ را بسته به پسندیده‌شدن یا پسندیده نشدن اختیار کند.

برای مثال اگر عنصر (i, j) مقدار یک را داشته باشد به این معنی است که کاربر سطر i آهنگ ستون j را پسندیده است.

مثال ماتریس آهنگ‌ها:

	Song 1	Song 2	Song 3	Song 4
Ali	1	0	1	0
Amin	1	1	0	0
Mohsen	1	0	1	1
Arman	1	0	0	0
Saeed	1	0	1	1
Bardia	1	0	1	0

○ ماتریس شباهت کاربران

در این مرحله بر اساس اطلاعات بخش قبل ماتریس جدیدی ایجاد می‌کنیم که سطرها و ستون‌های آن کاربران هستند و هر عنصر نشان‌دهنده شباهت بین دو کاربر است که عددی اعشاری از ۰ تا ۱ میتواند باشد که ۱ نشان‌دهنده بیشترین میزان شباهت است.

شباهت بر اساس تعداد آهنگ‌های پسندیده‌شده‌ی مشترک دو کاربر، تقسیم بر تعداد کل آهنگ‌ها حساب می‌شود.

برای مثال اگر عنصر (i, j) مقدار ۰.۶۳ را داشته باشد به این معنی است که کاربر سطر i ام و کاربر سطر j ام ۶۳ درصد شباهت دارند.

○ پیشنهاد آهنگ

در این بخش با استفاده از شباهت‌هایی که در مرحله قبل حساب کردیم سعی می‌کنیم به هر کاربر آهنگ‌های دیگری که ممکن است دوست داشته باشد را پیشنهاد بدهیم.

معیاری که به دنبال آن هستیم را **confidence** می‌نامیم که از مجموع شباهت‌ها ضرب در وضعیت پسندیده شدن آهنگ به دست می‌آید.

$$confidence(user, song) = \frac{\sum_{user_i} has_liked(user_i, song) * similarity(user, user_i)}{other\ users\ count}$$

برای مثال اگر بخواهیم آهنگ‌هایی که علی احتمالاً دوست خواهد داشت را به دست بیاوریم، به ازای هر آهنگ که علی **نپسندیده** است **confidence** بین سایر کاربران را حساب می‌کنیم و در نهایت بعد از حساب کردن این عدد برای هر آهنگ که علی **نپسندیده**، آهنگ‌هایی که مقدار **confidence** بیشتری برای آن‌ها محاسبه شده‌است را به علی پیشنهاد می‌دهیم.

مثال ماتریس شباهت‌ها:

	Ali	Amin	Mohsen	Arman	Saeed	Bardia
Ali		$\frac{1}{4}$	$\frac{2}{4}$	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{2}{4}$
Amin	$\frac{1}{4}$		$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
Mohsen	$\frac{2}{4}$	$\frac{1}{4}$		$\frac{1}{4}$	$\frac{3}{4}$	$\frac{2}{4}$
Arman	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$		$\frac{1}{4}$	$\frac{1}{4}$
Saeed	$\frac{2}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$		$\frac{2}{4}$
Bardia	$\frac{2}{4}$	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{1}{4}$	$\frac{2}{4}$	

محاسبه آهنگ‌های پیشنهادی به علی:

Song2 confidence score for Ali: $\frac{1 \times \frac{1}{4} + 0 \times \frac{2}{4} + 0 \times \frac{1}{4} + 0 \times \frac{2}{4} + 0 \times \frac{2}{4}}{5} = 0.05$

Song4 confidence score for Ali: $\frac{0 \times \frac{1}{4} + 1 \times \frac{2}{4} + 0 \times \frac{1}{4} + 1 \times \frac{2}{4} + 0 \times \frac{2}{4}}{5} = 0.2$

مشاهده می‌شود که به ترتیب آهنگ ۴ و آهنگ ۲ برای پیشنهاد به علی مناسب هستند.

• دستورات جدید

○ شبیه‌ترین افراد

کاربر پس از ورود به سامانه، می‌تواند با استفاده از این دستور نام کاربری و درصد شباهت تعداد مشخصی از کاربران را براساس آهنگ‌هایی که پسندیده‌اند، به ترتیب شباهت در خروجی مشاهده کند.

در صورتی که عدد count وارد شده از تعداد کل کاربران غیر از خود کاربر بیشتر باشد، همه کاربران دیگر به ترتیب شباهت در خروجی چاپ می‌شوند. در صورتی که شباهت برابر بود باید بر اساس ترتیب الفبایی نام کاربری چاپ بشود.

دقت داشته باشید که درصدها را با دقت دو رقم اعشار و بدون رند کردن و صرفاً با حذف کردن بخش بعد از دو رقم اعشار چاپ کنید. برای این کار می‌توانید از تابع `setprecision` استفاده کنید.

ورودی	خروجی
GET similar_users ? count <count>	<similarity>% <username1> <similarity>% <username2> <similarity>% <username3>... Bad Request Permission Denied
ورودی نمونه	خروجی نمونه
GET similar_users ? count 3	93.65% Asghar 82.40% Akbar 21.00% Mohammad

○ آهنگ‌های پیشنهادی

کاربر با استفاده از این دستور می‌تواند تعداد مشخصی آهنگ پیشنهادی را بر اساس روشی که بالاتر توضیح داده شد ببیند. اگر count بیشتر از تعداد آهنگ‌هایی که لایک نکرده باشد بایستی به کاربر تمام آهنگ‌های دیگر به ترتیب نمایش داده شود.

دقت داشته باشید که خروجی در این بخش بایستی به ترتیب مقدار میانگین وزن دار محاسبه شده چاپ شود و درصد confidence که چاپ می‌شود، همین مقدار است. در صورتی که این مقدار برابر بود باید به ترتیب شناسه آهنگ چاپ بشود.

این درصدها را نیز با دقت دو رقم اعشار نشان دهید.

ورودی	خروجی
GET recommended ? count <count>	<id> <confidence>% <title> <artist> <release_year> Bad Request Permission Denied
ورودی نمونه	خروجی نمونه
GET recommended ? count 3	3 98.20% Shape Of My Heart Sting 1993 2 82.00% Morghe Sahar Mohammadreza Shajarian 1990 1 79.54% Perfect Ed Sheeran 2017

• خطاها

○ خطای عدم وجود

اگر دسترسی که کاربر وارد می کند (قسمت پس از POST یا GET یا DELETE)، در لیست دستورهای تعریف شده نباشد یا کاربر با وارد کردن شناسه‌ی ناموجود در سیستم قصد دسترسی به موجودیتی را داشته باشد، این خطا در خروجی چاپ می شود.

خروجی
Not Found

○ خطای درخواست اشتباه

اگر اولین قسمت خط ورودی کاربر یکی از سه مقدار GET، POST و DELETE نباشد یا این دستور اطلاعات کافی برای اجرا را در خود نداشته باشد، سیستم خروجی زیر را چاپ می کند.

خروجی
Bad Request

○ خطای عدم دسترسی

اگر کاربری به سامانه وارد نشده باشد و از دستورات غیر از ثبت نام یا ورود استفاده کند، سیستم باید خروجی زیر را به کاربر نشان دهد.

خروجی
Permission Denied

نحوه‌ی تحویل و نکات

- تمام فایل‌های خود را در قالب یک پرونده‌ی زیپ با نام A7-<SID>.zip در صفحه‌ی Elearn درس بارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۸۹۹۹ است، نام پرونده‌ی شما باید A7-810198999.zip باشد.

○ برای مثال، نمونه فایل مورد قبول در زیر آمده است:

A7-810198999.zip

```
|_ ...  
|_ makefile
```

- بهتر است تا طراحی برنامه‌ی شما طوری باشد که کمترین وابستگی میان منطق برنامه و رابط کاربری آن وجود داشته باشد.
- توجه داشته باشید که حالت‌های خاصی که در صورت پروژه ذکر نشده است در تست‌های خودکار نخواهد بود و هر طور خواستید می‌توانید آن‌ها را مدیریت کنید.
- **دقت کنید** که پرونده زیپ آپلودی شما باید پس از Unzip شدن شامل پرونده‌های پروژه شما (از جمله Makefile) باشد و از زیپ کردن پوشه‌ای که داخل آن فایل‌های پروژه‌تان قرار دارد خودداری فرمایید.
- برنامه‌ی شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- **دقت کنید** که پروژه شما باید Multi-file باشد و Makefile داشته باشد. همین‌طور در Makefile خود مشخص کنید که از استاندارد c++11 استفاده می‌کنید.
- درستی برنامه‌ی شما از طریق **آزمون‌های خودکار** سنجیده می‌شود؛ بنابراین پیشنهاد می‌شود با استفاده از ابزارهایی مانند diff خروجی برنامه خود را با خروجی‌هایی که در اختیاران قرار داده شده است مطابقت دهید.
- **دقت کنید** که نام پرونده‌ی اجرایی شما باید **utunes.out** باشد.
- طراحی درست، رعایت سبک برنامه نویسی درست و تمیز بودن کد برنامه‌ی شما در نمره‌ی تمرین تأثیر زیادی دارد.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.
- سوالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن‌ها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این فاز پروژه ارتباط برقرار کنید.